Everglades Telemetry Overview

Contents

[Telemetry Output 2](#_Toc535396528)

[Telem\_GAME\_Scores 2](#_Toc535396529)

[Telem\_GROUP\_CombatUpdate 2](#_Toc535396530)

[Telem\_GROUP\_CreateNew 2](#_Toc535396531)

[Telem\_GROUP\_Disband 2](#_Toc535396532)

[Telem\_GROUP\_Initialization 3](#_Toc535396533)

[Telem\_GROUP\_Knowledge 3](#_Toc535396534)

[Telem\_GROUP\_MoveUpdate 3](#_Toc535396535)

[Telem\_GROUP\_TransferUnits 4](#_Toc535396536)

[Telem\_NODE\_ControlUpdate 4](#_Toc535396537)

[Telem\_NODE\_Knowledge 4](#_Toc535396538)

[Telem\_PLAYER\_Tags 4](#_Toc535396539)

[AI Location 5](#_Toc535396540)

[Known Issues 6](#_Toc535396541)

# Telemetry Output

After running a match, the telemetry is output in two locations, *C:\Everglades\server\Telemetry* and *C:\Everglades\EvergladesClient\EvergladesClient\Telemetry*. Additionally, in the *EvergladesClient* telemetry folder, a CSV is generated for every group of runs. This .csv is created with the file name **<Timestamp>\_<BluePlayerScript>\_<RedPlayerScript>.csv** that collates all the results into a single file. The csv header layout of this file is: **p0script\_blue,p1script\_red,p1score,p2score,wintype**

Possible Win Types:

* 1: Time Up
* 2: Base Capture
* 3: No More Units on either side (Mutually Assured Destruction)

The telemetry folders themselves consist of a series of folders containing .csv’s for different event types. The folders and events are detailed below.

### Telem\_GAME\_Scores

This event occurs every turn to update the HUD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Turn Number | Player1 | Player2 | Status | Focus |
| On which turn did this event take place? | The score for the blue player | The score for the red player | The win type of the match (See above) | The Group that the Camera AI has determined should be the focus. |

### Telem\_GROUP\_CombatUpdate

This event occurs on every combat update

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Turn Number | Player | Node | Groups | Units | Health |
| On which turn did this event take place? | The player involved in this combat update | The node where this combat event is taking place | A parallel array of GroupID’s | A parallel array of Unit ID’s | A parallel array of remaining health |

### Telem\_GROUP\_CreateNew

This event is called whenever a group splits into two. It will usually be followed by a TransferUnits event.

|  |  |  |  |
| --- | --- | --- | --- |
| Turn Number | Player | Node | Group |
| On which turn did this event take place? | The player that is making a new group | The Node where this group is created | The new GroupID used in future events referencing this group. |

### Telem\_GROUP\_Disband

This event occurs whenever a group is removed, mostly due to combat.

|  |  |  |
| --- | --- | --- |
| Turn Number | Player | Group |
| On which turn did this event take place? | The player this event pertains to. | The group that is disbanding. |

### Telem\_GROUP\_Initialization

This event only occurs at the beginning of a match. Turn 0 is always each team’s setup turn.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Turn Number | Player | Group | Node | Types | Start | Count |
| On which turn did this event take place? | The player we are initializing | The GroupID we are defining | The ID of the node where this group is spawned | An array of types that will be created | The start UnitID of each set of types | The amount of each type to create, parallel to the other two arrays. |

### Telem\_GROUP\_Knowledge

This event reflects the knowledge of the player of nodes they have access to.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Turn Number | Player | unitTypes | unitCount | Status | Node1 | Node2 |
| On which turn did this event take place? | The Player this event pertains to | List of unit types in this group | List of unit counts in this group | 0 for at node, 1 for in transit between nodes | If group is at node, this is the ID. If they are in transit this is the start node ID | If the group is in transit, this is the destination node ID. Otherwise, unused. |

### Telem\_GROUP\_MoveUpdate

This event is called whenever a group moves from one node to another.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Turn Number | Player | Group | Start | Destination | Status |
| On which turn did this event take place? | The player moving | The group that is moving | The NodeID where this move command started | The NodeID of the destination node | The status of the movement: RDY\_TO\_MOVE indicates moving to the edge, IN\_TRANSIT is during spline transition between nodes, ARRIVED when spline transition is complete |

### Telem\_GROUP\_TransferUnits

This event facilitates moving units from one group to another.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Turn Number | Player | Node | Group | Units |
| On which turn did this event take place? | The player that is involved in this event. | The node where this event happens | The group that will receive the new units | A list of unit IDs to move. These units will be removed from their existing group |

### Telem\_NODE\_ControlUpdate

This event occurs whenever there is an update to the control value of a node.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Turn Number | Node | Faction | controlValue | controlled |
| On which turn did this event take place? | The Node where an update to the control status occurred | The Player ID of the controlling player | The current control value of the node | A bool defining whether the node has been claimed. |

### Telem\_NODE\_Knowledge

This event shows the current knowledge of each player on each node.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Turn Number | Player | Nodes | Knowledge | Controller | Percent |
| On which turn did this event take place? | The player id that receives the message | List of node ids | List of node knowledge (0 = none, 1 = partial, 2 = full) | List of IDs representing who controls the node (-1 is neutral) | List of percentage of control for the node. |

### Telem\_PLAYER\_Tags

This event simply documents the AI used in this match.

|  |  |  |
| --- | --- | --- |
| Turn Number | Player 1 | Player 2 |
| This turn number is the last turn in the match. | The script used for Player 1 | The script used for Player 2 |

# AI Location

The AI files are in the folder *C:\Everglades\server\SourcePython.* We’ve included a few basic AI files:

* everglades\_stallAndSneak.py
* everglades\_veryAggressive.py
* everglades\_veryPassive.py
* everglades\_scoutAndDefend.py
* everglades\_sample.py

New AI files should be prefixed with “**everglades\_**” in order to show up in the drop down in the Everglades Client.

## Important Files

The AI file **everglades\_scoutAndDefend.py** is the most developed AI script of the bunch.

The file **everglades\_template.py** can be used as a blank template for creating new AI’s.

The file **evgcommands.py** is a self-documented set of commands that can be used in the AI.

The file **evgtypes.py** defines the data types used in the AI scripts.

# Known Issues

* When using Cam AI to follow the action, the camera can sometimes move through terrain.
* We have seen, on clean machines, the Python ZMQ package not properly installing. If the server seems to be throwing errors, please attempt the following commands to clean and reinstall pyZMQ:
  + python -m pip install –upgrade pip
  + pip uninstall zmq
  + pip uninstall pyZMQ
  + pip install PyHamcrest
  + pip install pyZMQ